1 Based on what you have learnt in the class, do the following steps:

a. Create a new folder

b. Put the following files in the folder

Code.txt

Log.txt

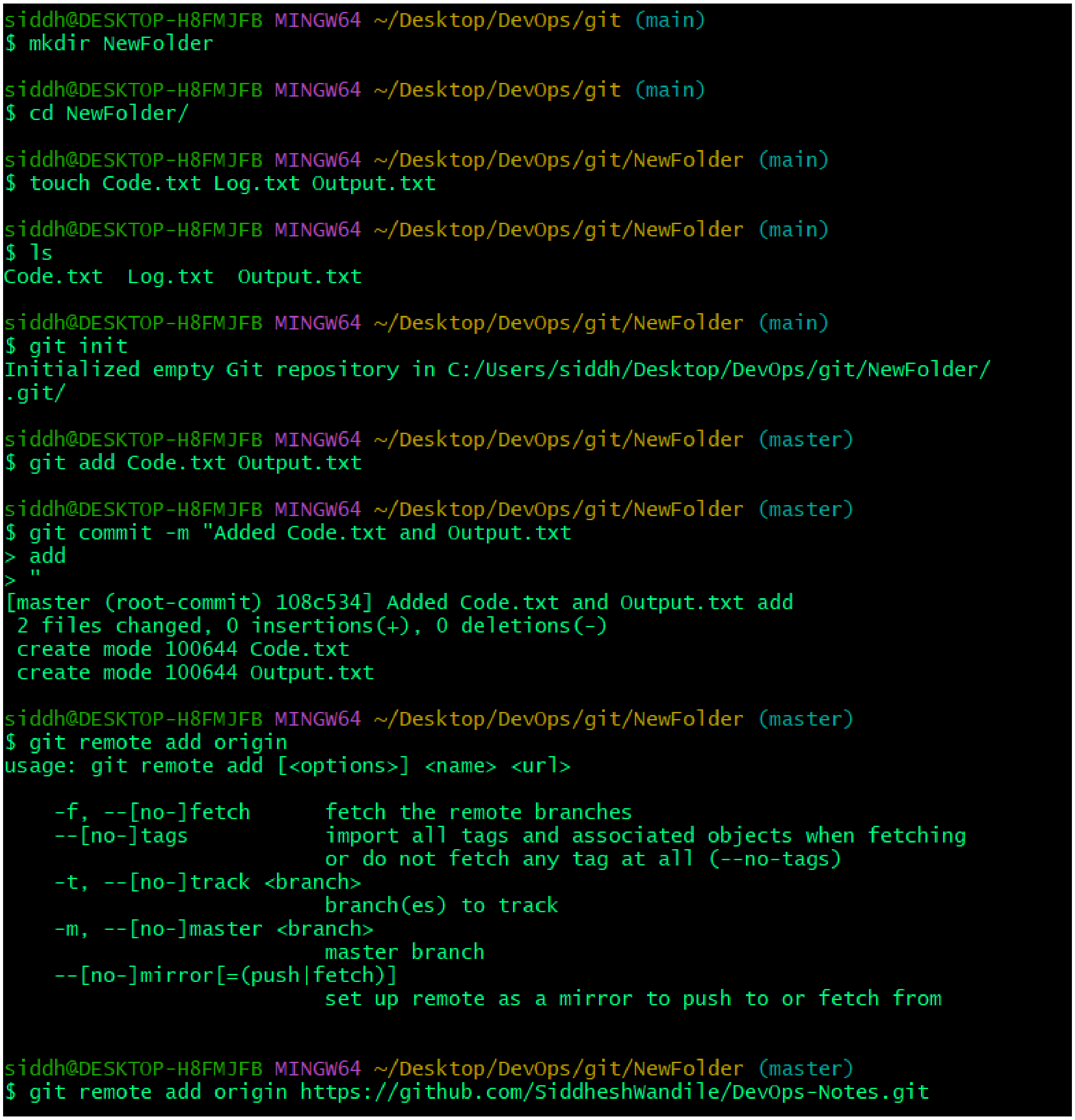
Output.txt

c. Stage the Code.txt and Output.txt files

d. Commit them

e. And finally push them to GitHub

=>



```
siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git (main)
$ mkdir NewFolder

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git (main)
$ cd NewFolder/

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (main)
$ touch Code.txt Log.txt Output.txt

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (main)
$ ls
Code.txt  Log.txt  Output.txt

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (main)
$ git init
Initialized empty Git repository in C:/Users/siddh/Desktop/DevOps/git/NewFolder/
.git/

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$ git add Code.txt Output.txt

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$ git commit -m "Added Code.txt and Output.txt
> add
> "
[master (root-commit) 108c534] Added Code.txt and Output.txt add
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 Code.txt
 create mode 100644 Output.txt

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$ git remote add origin
usage: git remote add [<options>] <name> <url>

    -f, --[no-]fetch      fetch the remote branches
    --[no-]tags           import all tags and associated objects when fetching
                          or do not fetch any tag at all (--no-tags)
    -t, --[no-]track <branch>
                          branch(es) to track
    -m, --[no-]master <branch>
                          master branch
    --[no-]mirror[=(push|fetch)]
                          set up remote as a mirror to push to or fetch from


siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$ git remote add origin https://github.com/SiddheshWandile/DevOps-Notes.git
```

**2. Tasks to Be Performed:**

1. Create a Git working directory with feature1.txt and feature2.txt in the master branch

2. Create 3 branches develop, feature1 and feature2

3. In develop branch create develop.txt, do not stage or commit it

4. Stash this file and check out to feature1 branch

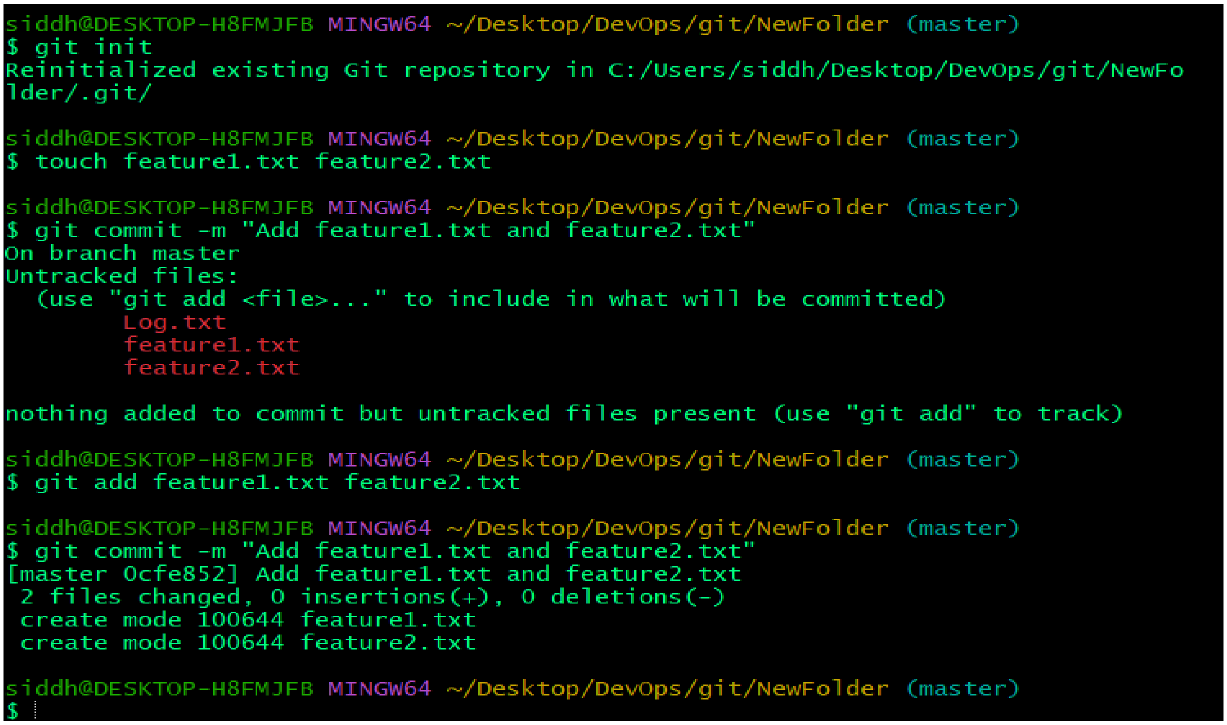
5. Create new.txt file in feature1 branch, stage and commit this file

6. Checkout to develop, unstash this file and commit

7. Please submit all the Git commands used to do the above step

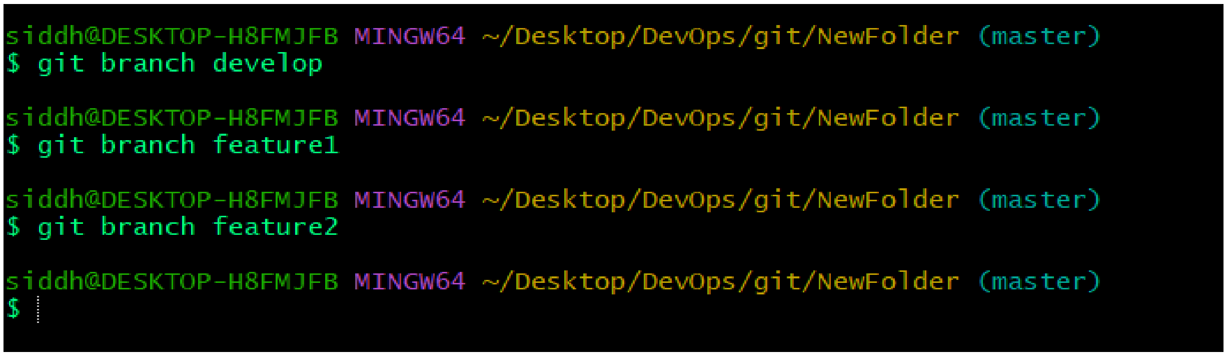
=>

1.



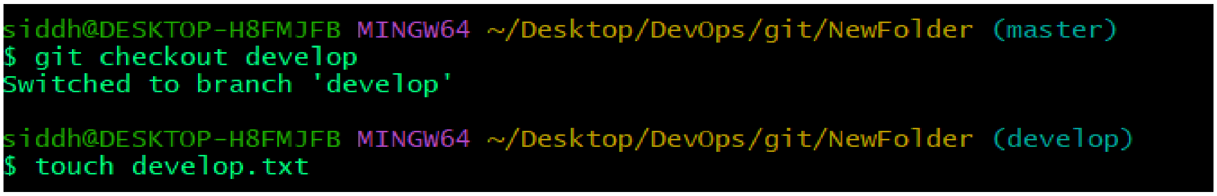
```
siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$ git init
Reinitialized existing Git repository in C:/Users/siddh/Desktop/DevOps/git/NewFo
lder/.git/

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$ touch feature1.txt feature2.txt

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$ git commit -m "Add feature1.txt and feature2.txt"
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Log.txt
        feature1.txt
        feature2.txt

nothing added to commit but untracked files present (use "git add" to track)

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$ git add feature1.txt feature2.txt

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$ git commit -m "Add feature1.txt and feature2.txt"
[master 0cfe852] Add feature1.txt and feature2.txt
 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 feature1.txt
 create mode 100644 feature2.txt

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$
```

2.



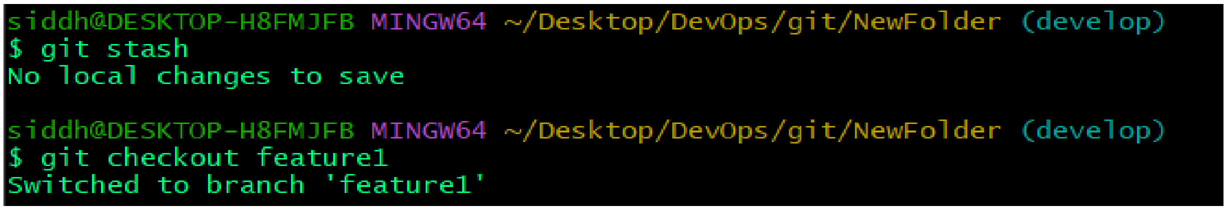
```
siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$ git branch develop

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$ git branch feature1

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$ git branch feature2

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$
```

3.



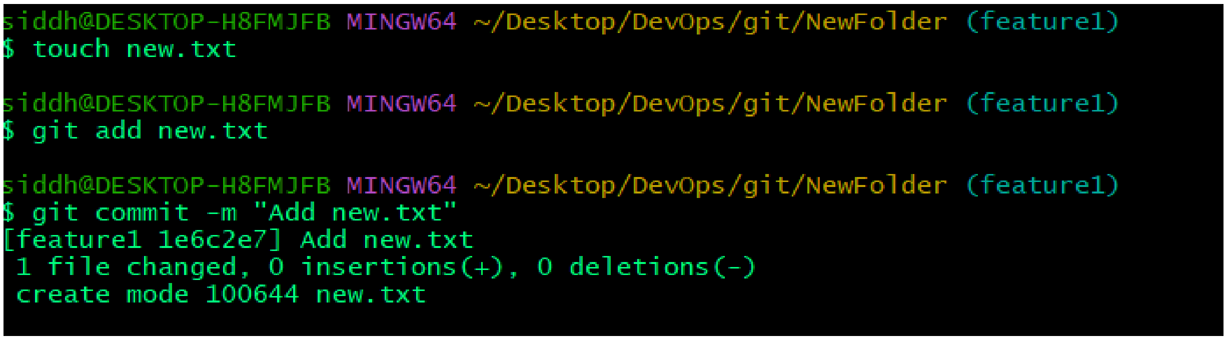
```
siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (master)
$ git checkout develop
Switched to branch 'develop'

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (develop)
$ touch develop.txt
```

4.



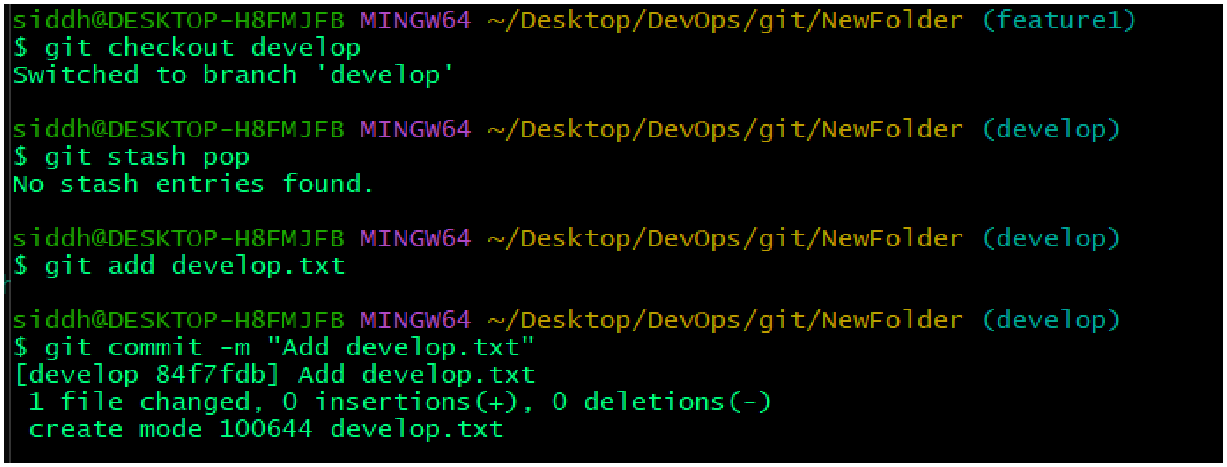
```
siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (develop)
$ git stash
No local changes to save

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (develop)
$ git checkout feature1
Switched to branch 'feature1'
```

5.



```
siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (feature1)
$ touch new.txt

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (feature1)
$ git add new.txt

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (feature1)
$ git commit -m "Add new.txt"
[feature1 1e6c2e7] Add new.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 new.txt
```

6.



```
siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (feature1)
$ git checkout develop
Switched to branch 'develop'

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (develop)
$ git stash pop
No stash entries found.

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (develop)
$ git add develop.txt

siddh@DESKTOP-H8FMJFB MINGW64 ~/Desktop/DevOps/git/NewFolder (develop)
$ git commit -m "Add develop.txt"
[develop 84f7fdb] Add develop.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 develop.txt
```

**3. Tasks to Be Performed:**

1. Create a Git working directory, with the following branches:

Develop

F1

f2

2. In the master branch, commit main.txt file

3. Put develop.txt in develop branch, f1.txt and f2.txt in f1 and f2 respectively

4. Push all these branches to GitHub

5. On local delete f2 branch

6. Delete the same branch on GitHub as well

=> **Step 1**

```
git init
git branch develop
git branch f1
git branch f2
```

**Step 2**

```
touch main.txt
git add main.txt
git commit -m "Add main.txt in the master branch"
```

**Step 3**

```
git checkout develop
touch develop.txt
git add develop.txt
git commit -m "Add develop.txt in develop branch"
git checkout f1
touch f1.txt
git add f1.txt
git commit -m "Add f1.txt in f1 branch"
git checkout f2
touch f2.txt
git add f2.txt
git commit -m "Add f2.txt in f2 branch"
```

**Step 4**

```
git remote add origin https://github.com/SiddheshWandile/DevOps-Notes.git
git push -u origin master
git push -u origin develop
git push -u origin f1
git push -u origin f2
```

**Step 5**

```
git branch -d f2
```

**Step 6**

```
git push origin --delete f2
```

**4. Tasks to Be Performed:**

1. Put master.txt on master branch, stage and commit

2. Create 3 branches: public 1, public 2 and private

3. Put public1.txt on public 1 branch, stage and commit

4. Merge public 1 on master branch

5. Merge public 2 on master branch

6. Edit master.txt on private branch, stage and commit

7. Now update branch public 1 and public 2 with new master code in private

8. Also update new master code on master

9. Finally update all the code on the private branch

=> **Step 1**

```
git init
touch master.txt
git add master.txt
git commit -m "Add master.txt on master branch"
```

**Step 2**

```
git branch public1
git branch public2
git branch private
```

**Step 3**

```
git checkout public1
touch public1.txt
git add public1.txt
git commit -m "Add public1.txt on public1 branch"
```

**Step 4**

```
git checkout master
git merge public1 -m "Merge public1 into master"
```

**Step 5**

```
git checkout public2
touch public2.txt
git add public2.txt
git commit -m "Add public2.txt on public2 branch"
git checkout master
git merge public2 -m "Merge public2 into master"
```

**Step 6**

```
git checkout private
echo "Updated content for master.txt" >> master.txt
git add master.txt
git commit -m "Edit master.txt on private branch"
```

**Step 7**

```
git checkout public1
git merge private -m "Update public1 with private branch changes"
git checkout public2
git merge private -m "Update public2 with private branch changes"
```

**Step 8**

```
git checkout master
git merge private -m "Update master with private branch changes"
```

**Step 9**

```
git checkout private
git merge master -m "Update private branch with new master code"
```

**5. Tasks to Be Performed**:

1. Create a Git Flow workflow architecture on Git
2. Create all the required branches
3. starting from e feature branch, push the branch to the master, following the architecture
4. Push an urgent.txt on master using hotfix